

# Package: sptotal (via r-universe)

September 8, 2024

**Type** Package

**Title** Predicting Totals and Weighted Sums from Spatial Data

**Version** 1.0.1

**Depends** R (>= 3.5.0)

**Description** Performs predictions of totals and weighted sums, or finite population block kriging, on spatial data using the methods in Ver Hoef (2008) <[doi:10.1007/s10651-007-0035-y](https://doi.org/10.1007/s10651-007-0035-y)>. The primary outputs are an estimate of the total, mean, or weighted sum in the region, an estimated prediction variance, and a plot of the predicted and observed values. This is useful primarily to users with ecological data that are counts or densities measured on some sites in a finite area of interest. Spatial prediction for the total count or average density in the entire region can then be done using the functions in this package.

**License** GPL (>= 2)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.1

**BugReports** <https://github.com/highamm/sptotal/issues>

**Imports** ggplot2, stats

**Suggests** rmarkdown, knitr, testthat (>= 3.0.0), prettydoc, tibble, gstat, graphics

**VignetteBuilder** knitr, rmarkdown

**URL** <https://highamm.github.io/sptotal/index.html>

**Config/testthat/edition** 3

**Repository** <https://highamm.r-universe.dev>

**RemoteUrl** <https://github.com/highamm/sptotal>

**RemoteRef** HEAD

**RemoteSha** 923eddce5689aa4c0e04576882af2fc5b2ac899b

## Contents

AIC.slmfit	2
AKmoose_df	3
check.variogram-deprecated	4
coef.slmfit	5
corModelExponential	5
exampledataset	6
fitted.slmfit	7
geostatSim	7
get.predinfo-deprecated	8
get.predplot-deprecated	9
GR2	10
LLtoTM	11
loglik.slmfit	12
m2LL.FPBK.nodet	12
mginv	13
plot.predict.slmfit	14
plot.slmfit	14
pointSimCSR	15
pointSimSyst	16
predict.slmfit	17
predict.stratafit	18
print.predict.slmfit	19
print.predict.stratafit	20
print.slmfit	20
print.summary.slmfit	21
residuals.slmfit	22
simdata	23
slmfit	24
sptotal	26
stratafit	26
summary.slmfit	28
summary.stratafit	29
sv	30
USlakes	31
<b>Index</b>	<b>32</b>

---

AIC.slmfit

---

*Extract the AIC from a `slmfit` object for comparing models.*


---

### Description

Extract the AIC from a `slmfit` object for comparing models.

**Usage**

```
## S3 method for class 'slmfit'
AIC(object, ...)
```

**Arguments**

```
object      a slmfit object
...         further arguments passed to or from other methods.
```

**Value**

The AIC value of the *slmfit* object. Here, AIC is computed as 2 times the negative log-likelihood plus 2 times the number of model parameters for Maximum Likelihood (ML) and 2 times the negative log-likelihood for REML. For REML, AIC should only be used to compare two models with the same covariates but different spatial covariance structures.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
AIC(slmobj)
```

---

AKmoose\_df

*Data Set with Alaska Moose Counts.*


---

**Description**

A data set that can be used with the *sptotal* package. In this example, the counts are of moose on 860 sites of equal area.

**Usage**

```
AKmoose_df
```

**Format**

A dataframe object. The data frame *AKmoose\_df* contains 860 rows and 8 columns:

**elev\_mean** The mean elevation for each site for each site  
**strat** A stratification variable (either L or M)  
**surveyed** Assigned a 1 if the site was surveyed and a 0 otherwise  
**total** The total moose count on each site (NA if the site was not surveyed)  
**x** the x-coordinate centroid of the site (TM)  
**y** the y-coordinate centroid of the site (TM)  
**lon** the longitudinal centroid of the site  
**lat** the latitudinal centroid of the site

## Source

[Alaska Department of Fish and Game, Division of Wildlife Conservation](#) has released this data set under the CC0 (creative commons) license. To the extent possible under law, Alaska Department of Fish and Game, Division of Wildlife Conservation waives all copyright and related or neighboring rights to An Alaskan GSPE (Geospatial Population Estimator) Survey of Moose, AKmoose\_df.rda. This work is published from: United States.

## Examples

```
data(AKmoose_df)
names(AKmoose_df)
summary(AKmoose_df)
```

---

check.variogram-deprecated

*Plot an Empirical Semi-Variogram of Residuals*

---

## Description

check.variogram() has been deprecated: use `plot.slmfit()` instead. Plots an empirical semi-variogram of the residuals from the spatial linear model with the fitted parametric model as a curve overtop of the binned points. By default, the empirical semi-variogram only shows distances that are less than or equal to the maximum distance in the data set divided by 2. Therefore, it's possible that the REML-fitted model will not "fit" the points perfectly.

## Usage

```
check.variogram(object)
```

## Arguments

object is an object of class `slmfit`.

## Value

a plot of the empirical semi-variogram with the fitted model overlaid.

## Examples

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
## Not Run
## check.variogram(slmobj)
```

---

coef.slmfit	<i>Extract Model Coefficients from a <code>slmfit</code> object</i>
-------------	---

---

**Description**

Extract Model Coefficients from a `slmfit` object

**Usage**

```
## S3 method for class 'slmfit'  
coef(object, ...)
```

**Arguments**

object	a <code>slmfit</code> object
...	further arguments passed to or from other methods.

**Value**

a vector of fitted model coefficients.

**Examples**

```
data(exampledataset) ## load a toy data set  
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampledataset,  
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')  
coef(slmobj)
```

---

corModelExponential	<i>Spatial Correlation Models</i>
---------------------	-----------------------------------

---

**Description**

Note that, currently, only three of these models are implemented in the `sptotal` package: `corModelExponential()`, `corModelGaussian()`, and `corModelSpherical()`.

**Usage**

```
corModelExponential(distance.matrix, range)
```

```
corModelGaussian(distance.matrix, range)
```

```
corModelSpherical(distance.matrix, range)
```

**Arguments**

`distance.matrix`      The distance matrix for sampled sites

`range`                The range that determines how quickly covariance among sites tapers

**Value**

Correlation Matrix

**Functions**

- `corModelExponential()`: Exponential Correlation Structure
- `corModelGaussian()`: Gaussian Correlation Structure
- `corModelSpherical()`: Spherical Correlation Structure

---

exampledataset

*Data Set with Uncorrelated Poisson Counts.*

---

**Description**

A toy data set that can be used with the `sptotal` package. In this example, the true counts are actually uncorrelated, the covariates are generated as uniform random variables, and the sites fall on a regular grid.

**Usage**

exampledataset

**Format**

A data frame with 40 rows and 7 variables:

**counts** counts, with NA values for unsampled sites

**pred1** a possible predictor

**pred2** a second possible predictor

**xcoords** coordinates on the x-axis

**ycoords** coordinates on the y-axis

**dummyvar** an extra variable

**areavar** Variable for the area of each plot ...

---

fitted.slmfit	<i>Extract Fitted Values from an slmfit object.</i>
---------------	---

---

**Description**

Extract Fitted Values from an slmfit object.

**Usage**

```
## S3 method for class 'slmfit'  
fitted(object, ...)
```

**Arguments**

object            a slmfit object generated from the `slmfit()` function.  
...               further arguments passed to or from other methods.

**Value**

a vector of fitted values (estimated means)

**Examples**

```
data(exampldataset) ## load a toy data set  
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,  
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')  
fitted(slmobj)
```

---

geostatSim	<i>Simulate geostatistical data on set of given locations</i>
------------	---

---

**Description**

Spatially correlated data are simulated assuming a multivariate normal random error vector. For simplicity, only "Exponential" and "Spherical" simulation options are given here.

**Usage**

```
geostatSim(  
  loc.data,  
  xcol = "x",  
  ycol = "y",  
  parsil = 1,  
  range = 1,  
  nugget = 0,  
  minorp = 1,
```

```

    rotate = 90,
    extrap = NULL,
    CorModel = "Exponential"
  )

```

### Arguments

loc.data	data.frame with x- and y-coordinates of locations for simulated data
xcol	name of the column in loc.data with x-coordinates, default is "x"
ycol	name of the column loc.data with y-coordinates, default is "y"
parsil	partial sill of autocorrelation model, default = 1
range	range of autocorrelation model, default = 1
nugget	range of autocorrelation model, default = 0
minorp	proportion of range in x direction to that of y direction for unrotated anisotropic model, default = 1
rotate	rotation of anisotropic axes, default = 90
extrap	extra covariance parameter
CorModel	autocorrelation model, default = "Exponential". Other possibilities are "Spherical".

### Value

data.frame of three columns, the original location data appended with a 3rd column of simulated geostatistical data

### Author(s)

Jay Ver Hoef

### Examples

```

locations <- expand.grid(1:10, 1:10)
geostatSim(locations, xcol = "Var1", ycol = "Var2",
  parsil = 4, range = 20, nugget = 1, CorModel = "Exponential")

```

---

get.predinfo-deprecated

*Display basic summary information in a tabular form.*

---

### Description

get.predinfo() has been deprecated. Information given is now provided in the basic print of a [predict.slmfit\(\)](#) object. Creates a list of tables that shows the prediction, standard error, and confidence interval for the prediction, as well as some summary information about the sample.



**Usage**

```
get.predinfo(x, conf_level = 0.9)
```

**Arguments**

`x` the output of the `predict.slmfit()` function, of class `predict.slmfit`  
`conf_level` is the confidence level for a normal-based confidence interval (default = 0.90).

**Value**

a list of three tables, including

- `simptab`, which contains the prediction and its standard error,
- `confbounds`, which contains a confidence interval for the prediction, and
- `outptmat`, a table of sampling information, including the number of sites sampled, the total number of sites, the total observed response, and the observed average density (equal to the average response if all site areas are equal).

**Examples**

```
data(exampdataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampdataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predobj <- predict(slmobj)
## Not Run
## get.predinfo(predobj)
```

---

```
get.predplot-deprecated
```

*Create a default map from predictions*

---

**Description**

Creates a default map for the predictions of unobserved sites. Note that all predictions are stored in a data frame in the output of `predict.slmfit()`. Therefore, if a user would like to create his or her own plot, he or she can easily do so using this data frame.

**Usage**

```
get.predplot(x)
```

**Arguments**

`x` the output of the `predict.slmfit()` function, of class `predict.slmfit`

**Value**

a plot with x-coordinates on the x-axis and y-coordinates on the y-axis that is coloured by predictions, with points with an X denoting that a site was sampled and filled circles denoting unsampled sites.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predobj <- predict(slmobj)
## Not Run
## get.predplot(predobj)
```

---

GR2

*Computes the Generalized R-squared.*

---

**Description**

Computes the Generalized R-squared.

**Usage**

```
GR2(object)
```

**Arguments**

object is an object of class `slmfit`.

**Value**

the value for generalized R-squared

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
GR2(slmobj)
```

---

LLtoTM *Convert Lat and Long to Transverse Mercator (TM)*

---

### Description

Latitude and Longitude coordinates are converted to TM projection coordinates with a user-defined central meridian. The resulting units from applying the function are kilometers.

### Usage

```
LLtoTM(cm, lat, lon, xcol = "x", ycol = "y", minx = NULL, miny = NULL)
```

### Arguments

cm	is the user defined central median. A common choice is the mean of the longitude values in your data set
lat	is the vector of latitudes
lon	is the vector of longitudes
xcol	is the name of the output TM column of x coordinates
ycol	is the name of the output TM column of y coordinates
minx	is 'NULL' by default and sets the minimum x-coordinate value to 0. This is an optional minimum value for the x-coordinate vector.
miny	is 'NULL' by default and sets the minimum y-coordinate value to 0. This is an optional minimum value for the y-coordinate vector.

### Details

This function should only be used if the coordinates supplied by the user are latitude and longitude. The default TM projection here specifies that both the minimum x and y-coordinate values are 0 scaled to 1 km.

### Value

A list with the TM coordinates as the first component of the list. The first component of the list contains x coordinates in the first column and y coordinates in the second column. The remaining elements of the list are the cm, minx, and miny values that were input.

### Examples

```
## Add transverse Mercator x and y coordinates to a data frame with
## latitude/longitude coordinates.
exampledataset$xc_TM_ <- LLtoTM(cm = base::mean(exampledataset[, "xcoords"]),
  lat = exampledataset[, "ycoords"],
  lon = exampledataset[, "xcoords"])$xy[, 1]
exampledataset$yc_TM_ <- LLtoTM(cm = base::mean(exampledataset[, "xcoords"]),
  lat = exampledataset[, "ycoords"],
  lon = exampledataset[, "xcoords"])$xy[, 2]
```

---

loglik.slmfit	<i>Extract Log-Likelihood from a fitted class slmfit object</i>
---------------	---

---

**Description**

Extract Log-Likelihood from a fitted class `slmfit` object

**Usage**

```
loglik.slmfit(object, ...)
```

**Arguments**

<code>object</code>	is a class <code>slmfit</code> object
<code>...</code>	additional arguments

**Value**

the log-likelihood of the model fit in the `slmfit` object

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
loglik.slmfit(slmobj)
```

---

m2LL.FPBK.nodet	<i>Covariance Parameter Estimation Function.</i>
-----------------	--

---

**Description**

The primary purpose of `m2LL.FPBK.nodet()` is to estimate the spatial covariance parameters using REML. This is a helper function to `slmfit()`.

**Usage**

```
m2LL.FPBK.nodet(theta, zcol, XDesign, distmat, CorModel, estmethod)
```

**Arguments**

theta	is the parameter vector of (nugget, partialsill, range)
zcol	is the response vector of densities
XDesign	is the design matrix containing the covariates used to predict animal or plant abundance (including a column of 1's for the intercept).
distmat	is the distance matrix of the sampled sites
CorModel	is the geostatistical spatial correlation model to be used. See the corModels documentation for possible models to use.
estmethod	is either "REML" for restricted maximum likelihood or "ML" for maximum likelihood.

**Value**

A numeric output of minus 2 times the restricted log likelihood to be minimized by 'optim' to obtain spatial parameter estimates.

---

mginv

---

*Constructing the generalized inverse of a matrix*


---

**Description**

Computes the generalized inverse of a matrix X. This function is used in the m2LL.FPBK.nodet functions in order to estimate the spatial covariance parameters

**Usage**

```
mginv(X, tol = sqrt(.Machine$double.eps))
```

**Arguments**

X	The matrix to be inverted
tol	The tolerance of the estimation

**Value**

The generalized inverse matrix

---

```
plot.predict.slmfit
```

*Create a default map from predictions*

---

### Description

Creates a default map for the predictions of unobserved sites. Note that all predictions are stored in a data frame in the output of `predict.slmfit()`. Therefore, if a user would like to create his or her own plot, they can easily do so using this data frame.

### Usage

```
## S3 method for class 'predict.slmfit'
plot(x, ...)
```

### Arguments

`x` the output of the `predict.slmfit()` function, of class `predict.slmfit`  
`...` further arguments passed to or from other methods.

### Value

a plot with x-coordinates on the x-axis and y-coordinates on the y-axis that is coloured by predictions, with points with an X denoting that a site was sampled and filled circles denoting unsampled sites.

### Examples

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predobj <- predict(slmobj)
plot(predobj)
```

---

```
plot.slmfit
```

*Plot an Empirical Semi-Variogram of Residuals*

---

### Description

Plots an empirical semi-variogram of the residuals from the spatial linear model with the fitted parametric model as a curve overtop of the binned points. By default, the empirical semi-variogram only shows distances that are less than or equal to the maximum distance in the data set divided by 2. Therefore, it's possible that the REML-fitted model will not "fit" the points perfectly.

### Usage

```
## S3 method for class 'slmfit'
plot(x, ...)
```

**Arguments**

`x` is an object of class `slmfit`.  
`...` further arguments passed to or from other methods.

**Value**

a plot of the empirical semi-variogram with the fitted model overlaid.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
plot(slmobj)
```

```
data(exampldataset) ## load a toy data set
exampldataset$strata <- c(rep("A", 19), rep("B", 21))
strataobj <- slmfit(formula = counts ~ pred1 + pred2,
  data = exampldataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
plot(strataobj[[1]])
```

---

pointSimCSR

*simulate completely spatially random point patterns.*


---

**Description**

simulates a completely spatially random point patterns. This function is only used in simulating data sets.

**Usage**

```
pointSimCSR(
  npoints = 100,
  lower_x_lim = 0,
  upper_x_lim = 1,
  lower_y_lim = 0,
  upper_y_lim = 1
)
```

**Arguments**

`npoints` number of points to add that are completely spatially random (CSR), default = 100  
`lower_x_lim` left limit of boundary, default = 0  
`upper_x_lim` right limit of boundary, default = 1  
`lower_y_lim` lower limit of boundary, default = 0  
`upper_y_lim` upper limit of boundary, default = 1

**Value**

data.frame of two columns, x-coordinate in the first, and y-coordinate in the second.

**Author(s)**

Jay Ver Hoef

---

pointSimSyst

*Creates a systematic grid of points.*

---

**Description**

Creates a systematic grid of points. This function is only used in simulating data sets.

**Usage**

```
pointSimSyst(  
  nrow = 10,  
  ncol = 10,  
  lower_x_lim = 0,  
  upper_x_lim = 1,  
  lower_y_lim = 0,  
  upper_y_lim = 1  
)
```

**Arguments**

nrow	the number of rows in the systematic grid, default is 10
ncol	the number of cols in the systematic grid, default is 10
lower_x_lim	the lower limit for x-coordinate, default is 0
upper_x_lim	the upper limit for x-coordinate, default is 1
lower_y_lim	the lower limit for y-coordinate, default is 0
upper_y_lim	the upper limit for y-coordinate, default is 1

**Value**

A data.frame with x- and y-coordinates of simulated locations

**Author(s)**

Jay Ver Hoef



---

predict.slmfit      *Perform Finite Population Block Kriging*

---

### Description

Uses an object of class `slmfit` from the `slmfit()` function to predict the response on the unsampled sites. The column of the data set that has the response should have numeric values for the observed response on the sampled sites and 'NA' for any site that was not sampled. Note that there is no `newdata` argument to `predict.slmfit()`: any point in space for which a prediction is needed should be included in the original data set in `slmfit()` with the response variable as NA.

### Usage

```
## S3 method for class 'slmfit'
predict(object, wtscol = NULL, conf_level = 0.9, ...)
```

### Arguments

<code>object</code>	is an object generated from <code>slmfit()</code>
<code>wtscol</code>	is the name of the column that contains the weights for prediction. The default setting predicts the population total
<code>conf_level</code>	is the confidence level for a prediction interval, 0.90 by default
<code>...</code>	further arguments passed to or from other methods.

### Value

a list with

- the estimated population total
- the estimated prediction variance
- a data frame containing
  1. x-coordinates
  2. y-coordinates
  3. density predictions
  4. count predictions
  5. site-by-site density prediction variances
  6. site-by-site count prediction variances
  7. indicator variable for whether or not the each site was sampled
  8. estimated mean for each site
  9. area of each site
- vector with estimated covariance parameters
- the formula used to fit the model in `slmfit()`
- the confidence level
- the confidence interval bounds

**Examples**

```
data(exampledataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampledataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predict(slmobj)
```

---

predict.stratafit      *Perform Finite Population Block Kriging*

---

**Description**

Uses an object of class `stratafit` from the `stratafit()` function to predict the response on the unsampled sites for separate strata. The column of the data set that has the response should have numeric values for the observed response on the sampled sites and 'NA' for any site that was not sampled. Note that there is no `newdata` argument to `predict.stratafit()`: any point in space for which a prediction is needed should be included in the original data set in `stratafit()` with the response variable as NA.

**Usage**

```
## S3 method for class 'stratafit'
predict(object, wtscol = NULL, conf_level = 0.9, ...)
```

**Arguments**

<code>object</code>	is an object generated from <code>stratafit()</code>
<code>wtscol</code>	is the name of the column that contains the weights for prediction. The default setting predicts the population total
<code>conf_level</code>	by default, 0.90, this is the desired confidence level for a prediction interval
<code>...</code>	further arguments passed to or from other methods.

**Value**

a list with

- the estimated population total
- the estimated prediction variance
- a data frame containing
  1. x-coordinates
  2. y-coordinates
  3. density predictions
  4. count predictions
  5. site-by-site density prediction variances
  6. site-by-site count prediction variances
  7. indicator variable for whether or not the each site was sampled

- 8. estimated mean for each site
- 9. area of each site
- vector with estimated covariance parameters
- the formula used to fit the model in `slmfit()`

## Examples

```
data(exampledataset) ## load a toy data set
exampledataset$strata <- c(rep("A", 19), rep("B", 21))
strataobj <- stratafit(formula = counts ~ pred1 + pred2,
  data = exampledataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predict(strataobj)
```

---

`print.predict.slmfit` Prints a short summary for the `predict.slmfit()` function.

---

## Description

This function uses the object that is output from `predict.slmfit()` of class `predict.slmfit`.

## Usage

```
## S3 method for class 'predict.slmfit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

## Arguments

<code>x</code>	is a prediction object generated from <code>predict.slmfit()</code>
<code>digits</code>	is the number of digits to be displayed in the model output
<code>...</code>	further arguments passed to or from other methods.

## Examples

```
data(exampledataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampledataset,
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
print(predict(slmobj), digits = 4)
```

---

```
print.predict.stratafit
```

*Prints a short summary for the `predict.stratafit()` function.*

---

### Description

This function uses the object that is output from `predict.stratafit()` of class `predict.stratafit`.

### Usage

```
## S3 method for class 'predict.stratafit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

### Arguments

<code>x</code>	is a prediction object generated from <code>predict.stratafit()</code>
<code>digits</code>	is the number of digits to be displayed in the model output
<code>...</code>	further arguments passed to or from other methods.

### Value

a list with

- a table of predictions, standard errors, and confidence intervals for each stratum and for the total.

### Examples

```
data(exampledataset) ## load a toy data set
exampledataset$strata <- c(rep("A", 19), rep("B", 21))
strataobj <- stratafit(formula = counts ~ pred1 + pred2,
  data = exampledataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
predict(strataobj)
```

---

```
print.slmfit
```

*Prints the fitted coefficient table of a fitted spatial linear model.*

---

### Description

This function uses the object that is output from `slmfit()` of class `slmfit`.

### Usage

```
## S3 method for class 'slmfit'
print(x, digits = max(3L, getOption("digits") - 3L), ...)
```

**Arguments**

`x` is an object generated from `slmfit()`  
`digits` is the number of digits to be displayed in the model output  
`...` further arguments passed to or from other methods.

**Value**

a list with

- model formula
- summary statistics for the residuals.
- a table of fixed effects estimates and associated standard errors.
- estimated spatial covariance parameter estimates.
- generalized r-squared value.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
print(slmobj)
```

---

`print.summary.slmfit` *Prints the summary of a fitted spatial linear model.*

---

**Description**

This function uses the object that is output from `summary.slmfit()`.

**Usage**

```
## S3 method for class 'summary.slmfit'
print(
  x,
  digits = max(3L, getOption("digits") - 3L),
  signif.stars = getOption("show.signif.stars"),
  ...
)
```

**Arguments**

`x` is an summary object generated from `summary.slmfit()`  
`digits` is the number of digits to be displayed in the model output  
`signif.stars` is an option to show which predictors are significant.  
`...` further arguments passed to or from other methods.

**Value**

a list with

- model formula
- summary statistics for the residuals.
- a table of fixed effects estimates and associated standard errors.
- estimated spatial covariance parameter estimates.
- generalized r-squared value.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
print(summary(slmobj))
```

---

residuals.slmfit      *Extract Model Residuals from an slmfit object.*

---

**Description**

Extract Model Residuals from an slmfit object.

**Usage**

```
## S3 method for class 'slmfit'
residuals(object, type = "raw", cross.validation = FALSE, ...)
```

**Arguments**

**object**            a slmfit object generated from the `slmfit()` function.

**type**              residual type: "raw" (the default) or "normalized"

**cross.validation**   a logical (TRUE or FALSE) that indicates whether the residuals computed should be found using leave one out cross-validation. Set to FALSE by default.

**...**              further arguments passed to or from other methods.

**Value**

a vector of residuals, consisting of each observed response/density minus the estimated mean, or, in the case of cross-validation, the observed response/density minus the leave-one-out-cross-validation prediction.

## Examples

```
data(exampledataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampledataset,
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
residuals(slmobj)
residuals(slmobj, cross.validation = TRUE)
```

---

simdata

*Simulated Spatially Autocorrelated Data.*

---

## Description

A simulated data set that can be used with the `sptotal` package.

## Usage

```
simdata
```

## Format

A data frame object including:

**x** The x-coordinate for each site

**y** The y-coordinate for each site

**X1** Simulated independent variable to be used as a predictor

**X2** Simulated independent variable to be used as a predictor

**X3** Simulated independent variable to be used as a predictor

**X4** Simulated independent variable to be used as a predictor

**X5** Simulated independent variable to be used as a predictor

**X6** Simulated spatially correlated random variable to be used as a predictor

**X7** Simulated spatially correlated random variable to be used as a predictor

**F1** Simulated factor variable to be used as a predictor

**F2** Simulated factor variable to be used as a predictor

**Z** The simulated response variable.

**wts1** Prediction weights if estimating an overall mean

**wts2** Prediction weights for estimating a total over a subset of 25 contiguous plots

## Examples

```
data(simdata)
names(simdata)
summary(simdata)
```

slmfit

*Fits a Spatial Linear Model***Description**

Estimates regression coefficients and spatial autocorrelation parameters, given spatial coordinates and a model formula.

**Usage**

```
slmfit(
  formula,
  data,
  xcoordcol,
  ycoordcol,
  areacol = NULL,
  stratacol = NULL,
  CorModel = "Exponential",
  estmethod = "REML",
  covestimates = c(NA, NA, NA)
)
```

**Arguments**

formula	is an R linear model formula specifying the response variable as well as covariates for predicting the response on the unsampled sites.
data	is a data frame or tibble with the response column, the covariates to be used for the block kriging, and the spatial coordinates for all of the sites. Alternatively, data can be an sp Spatial Points Data Frame or sf object with POINT geometry.
xcoordcol	is the name of the column in the data frame with x coordinates or longitudinal coordinates
ycoordcol	is the name of the column in the data frame with y coordinates or latitudinal coordinates
areacol	is the name of the column with the areas of the sites. By default, we assume that all sites have equal area, in which case a vector of 1's is used as the areas.
stratacol	is the name of the the column with the stratification variable, if strata are to be fit separately, with different covariance parameter estimates.
CorModel	is the covariance structure. By default, CorModel is Exponential but other options include the Spherical and Gaussian.
estmethod	is either the default "REML" for restricted maximum likelihood to estimate the covariance parameters and regression coefficients or "ML" to estimate the covariance parameters and regression coefficients. This argument can also be set to "None", in which case covestimates must be provided.
covestimates	is an optional vector of covariance parameter estimates (nugget, partial sill, range). If these are given and estmethod = "None", the the provided vector are treated as the estimators to create the covariance structure.



**Value**

a list of class `slmfit` with

- the spatial covariance estimates
- the regression coefficient estimates
- the covariance matrix of the fixed effects
- minus two times the log-likelihood of the model
- the names of the predictors
- the sample size
- the name of the covariance model used
- a vector of residuals
- the design matrix
- a vector of the sampled densities
- a list containing
  1. formula, the model formula
  2. data, the data set input as the data argument
  3. xcoordcol, the name of the x-coordinate column
  4. ycoordcol, the name of the y-coordinate column
  5. estmethod, either REML or ML
  6. CorModel, the correlation model used
  7. estimated covariance matrix of all sites
  8. Inverted covariance matrix on the sampled sites
  9. the vector of areas.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
summary(slmobj)
```

```
data(exampldataset) ## load a toy data set
exampldataset$strata <- c(rep("A", 19), rep("B", 21))
strataobj <- slmfit(formula = counts ~ pred1 + pred2,
  data = exampldataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
summary(strataobj)
```

---

sptotal	<i>sptotal: A package used for performing Finite Population Block Kriging (FPBK) on polygonal count data.</i>
---------	---

---

### Description

The package provides an option to perform FPBK on counts assuming perfect detection of counts on the sites that were in the survey sample. The functions in the package use methods in (Ver Hoef, 2008, <doi:10.1007/s10651-007-0035-y>)

### Details

sptotal Main Functions:

`slmfit` fits a spatial linear model to the response on the observed/sampled sites.

`predict.slmfit` uses the spatial linear model fit from `slmfit` and finite population block kriging to predict the response at unobserved locations. A prediction for the total response as well as a prediction variance are given by default.

Most of the remaining functions in the package are either helper functions or extra optional functions to extract various specific things from an `slmfit` object, such as residuals, AIC, log-likelihood, etc.

See the Vignette for more details: `browseVignettes("sptotal")` Reference for Mathematical Details:

Ver Hoef, Jay M. "Spatial methods for plot-based sampling of wildlife populations." *A Environmental and Ecological Statistics* 15, no. 1 (2008): 3-13.

---

stratafit	<i>Fits a Separate Spatial Linear Model for Each Stratum</i>
-----------	--

---

### Description

Estimates regression coefficients and spatial autocorrelation parameters, given spatial coordinates, a model formula, and a stratification variable. Arguments are the same here as they are for `slmfit()`, with an extra argument for `stratacol`, the name of the stratification column. Note that stratum can either be incorporated as a covariate in `slmfit()`, in which case the errors have the same spatial covariance, or, models with differing spatial covariances for the errors can be fit to each level of stratum, as is done here in `stratafit()`.

### Usage

```
stratafit(
  formula,
  data,
  xcoordcol,
```

```

    ycoordcol,
    stratacol = NULL,
    areacol = NULL,
    CorModel = "Exponential",
    estmethod = "REML"
  )

```

### Arguments

formula	is an R linear model formula specifying the response variable as well as covariates for predicting the response on the unsampled sites.
data	is the data set with the response column, the covariates to be used for the block kriging, and the spatial coordinates for all of the sites.
xcoordcol	is the name of the column in the data frame with x coordinates or longitudinal coordinates
ycoordcol	is the name of the column in the data frame with y coordinates or latitudinal coordinates
stratacol	is the name of the stratification column
areacol	is the name of the column with the areas of the sites. By default, we assume that all sites have equal area, in which case a vector of 1's is used as the areas.
CorModel	is the covariance structure. By default, CorModel is Exponential but other options include the Spherical and Gaussian.
estmethod	is either the default "REML" for restricted maximum likelihood to estimate the covariance parameters and regression coefficients or "ML" to estimate the covariance parameters and regression coefficients.

### Value

a list of class `slmfit` with

- the spatial covariance estimates
- the regression coefficient estimates
- the covariance matrix of the fixed effects
- minus two times the log-likelihood of the model
- the names of the predictors
- the sample size
- the name of the covariance model used
- a vector of residuals
- the design matrix
- a vector of the sampled densities
- a list containing
  1. formula, the model formula
  2. data, the data set input as the data argument

3. xcoordcol, the name of the x-coordinate column
4. ycoordcol, the name of the y-coordinate column
5. estmethod, either REML or ML
6. CorModel, the correlation model used
7. estimated covariance matrix of all sites
8. Inverted covariance matrix on the sampled sites
9. the vector of areas.

### Examples

```
data(exampledataset) ## load a toy data set
exampledataset$strata <- c(rep("A", 19), rep("B", 21))
strataobj <- stratafit(formula = counts ~ pred1 + pred2,
  data = exampledataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
summary(strataobj)
```

---

summary.slmfit	<i>Summarizes a fitted spatial linear model.</i>
----------------	--

---

### Description

In conjunction with `print.summary.slmfit()`, the output looks similar to output from R's standard `lm()` function.

### Usage

```
## S3 method for class 'slmfit'
summary(object, ...)
```

### Arguments

object	is an object generated from <code>slmfit()</code> of class <code>slmfit</code> .
...	further arguments passed to or from other methods.

### Value

a list with

- model formula
- a table of fixed effects estimates and associated standard errors
- estimated spatial covariance parameter estimates
- residuals
- generalized r-squared.

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
summary(slmobj)
```

---

summary.stratafit	<i>Summarizes a fitted spatial linear model with a stratification variable..</i>
-------------------	--

---

**Description**

If a model is fitted with `stratafit()`, then this summary function produces summary output for each level of the stratification variable in the same style as the `summary.slmfit()` function.

**Usage**

```
## S3 method for class 'stratafit'
summary(object, ...)
```

**Arguments**

object	is an object generated from <code>slmfit()</code> of class <code>slmfit</code> .
...	further arguments passed to or from other methods.

**Value**

a list with

- model formula
- a table of fixed effects estimates and associated standard errors
- estimated spatial covariance parameter estimates
- residuals
- generalized r-squared.

**Examples**

```
data(exampldataset) ## load a toy data set
exampldataset$strata <- c(rep("A", 25), rep("B", 15))
strataobj <- stratafit(formula = counts ~ pred1 + pred2,
  data = exampldataset, stratacol = "strata",
  xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
summary(strataobj)
```

**Description**

Compute the empirical semivariogram for a data set with specified x and y spatial coordinates and residuals.

**Usage**

```
sv(data, xcoordcol, ycoordcol, residcol, bins = 15, cutoff = NULL, ...)
```

**Arguments**

<code>data</code>	A data frame or tibble that contains variables for the x coordinates, y coordinates, and residuals.
<code>xcoordcol</code>	is the name of the column in the data frame with x coordinates or longitudinal coordinates
<code>ycoordcol</code>	is the name of the column in the data frame with y coordinates or latitudinal coordinates
<code>residcol</code>	is the name of the column in the data frame with residuals.
<code>bins</code>	Number of equally spaced semivariogram bins. The default is 15.
<code>cutoff</code>	The maximum spatial distance to be considered. The default is half of the maximum observed distance in the data frame.
<code>...</code>	Additional arguments to be used by <code>stats::dist()</code> for spatial distance calculations.

**Value**

A data frame with the average distance in each bin (`dist`), the semivariance (`gamma`), and the number of unique pairs in each bin (`np`)

**Examples**

```
data(exampldataset) ## load a toy data set
slmobj <- slmfit(formula = counts ~ pred1 + pred2, data = exampldataset,
xcoordcol = 'xcoords', ycoordcol = 'ycoords', areacol = 'areavar')
sampleddataset <- na.omit(exampldataset)
svexample <- data.frame(
  xcoords = sampleddataset$xcoords,
  ycoords = sampleddataset$ycoords,
  resids = residuals(slmobj)
)
svdata <- sv(svexample, "xcoords", "ycoords", "resids")
```

---

USlakes

*Dissolved Organic Carbon in U.S. Lakes*

---

### Description

These data contain dissolved organic carbon (DOC) in National Lakes Data from the U.S. Environmental Protection Agency

### Usage

USlakes

### Format

A data frame with 1206 rows and 9 variables:

**XCOORD** x-coordinate from US Contiguous Albers Equal Area Conic projection

**YCOORD** y-coordinate from US Contiguous Albers Equal Area Conic projection

**DOC\_RESULT** Analyte value, in mg/L, for Dissolved Organic Carbon

**ELEVATION** Elevation at lake coordinates (LAT\_DD\_N83, LON\_DD\_N83) from NHD Digital Elevation Map layer

**FCIBIG\_LIT** Fish cover: index of fish cover due to large structures in the littoral zone

**RVFCGNDBARE\_RIP** riparian zone and vegetation: fraction of ground lacking cover in the riparian zone

**RVFCGNDWOODY\_RIP** riparian zone and vegetation: fraction of ground cover by woody vegetation in the riparian zone

**RVFPUNDWOODY\_RIP** riparian zone and vegetation: fraction of understory with nonwoody cover present in the riparian zone

**UID** A unique lake identifier in the EPA lake survey databases

### Source

[National Aquatic Resource Surveys](#) webpage. We combined [site data](#), [DOC data](#), and [habitat metrics](#) to create a data set of 1206 lakes in the conterminous United States.

### Examples

```
data(USlakes)
names(USlakes)
summary(USlakes)
```

# Index

## \* datasets

- AKmoose\_df, 3
  - exempldataset, 6
  - simdata, 23
  - USlakes, 31
- AIC.slmfit, 2
- AKmoose\_df, 3
- check.variogram  
(check.variogram-deprecated), 4
- check.variogram-deprecated, 4
- coef.slmfit, 5
- corModelExponential, 5
- corModelGaussian (corModelExponential),  
5
- corModelSpherical  
(corModelExponential), 5
- exempldataset, 6
- fitted.slmfit, 7
- geostatSim, 7
- get.predinfo (get.predinfo-deprecated),  
8
- get.predinfo-deprecated, 8
- get.predplot (get.predplot-deprecated),  
9
- get.predplot-deprecated, 9
- GR2, 10
- LLtoTM, 11
- loglik.slmfit, 12
- m2LL.FPBK.nodet, 12
- mginv, 13
- plot.predict.slmfit, 14
- plot.slmfit, 4, 14
- pointSimCSR, 15
- pointSimSyst, 16
- predict.slmfit, 8, 9, 14, 17, 19, 26
- predict.stratafit, 18, 20
- print.predict.slmfit, 19
- print.predict.stratafit, 20
- print.slmfit, 20
- print.summary.slmfit, 21
- residuals.slmfit, 22
- simdata, 23
- slmfit, 2–5, 7, 15, 17, 20–22, 24, 26, 28, 29
- sptotal, 26
- sptotal-package (sptotal), 26
- stratafit, 18, 26, 29
- summary.slmfit, 21, 28, 29
- summary.stratafit, 29
- sv, 30
- USlakes, 31